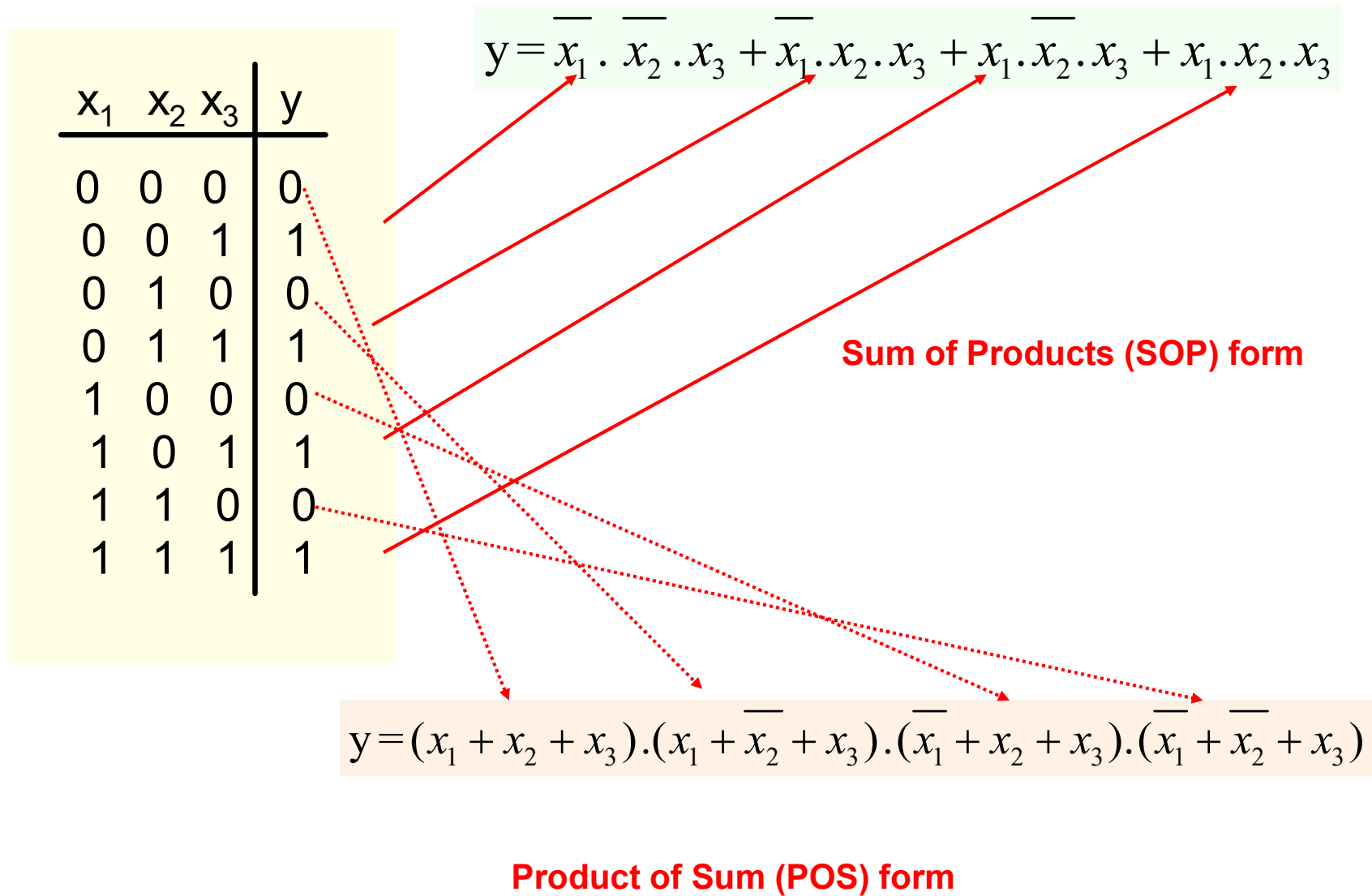


# ESC201T : Introduction to Electronics

## Lecture 33: Digital Circuits-3

B. Mazhari  
Dept. of EE, IIT Kanpur

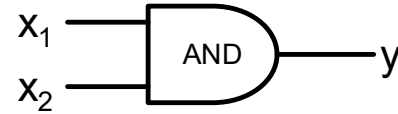
## Obtaining Boolean expressions from truth Table



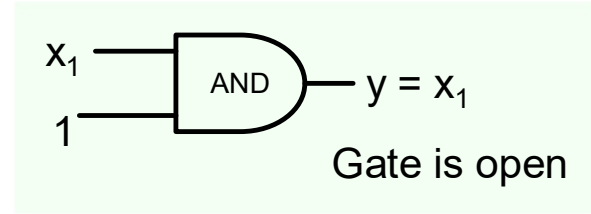
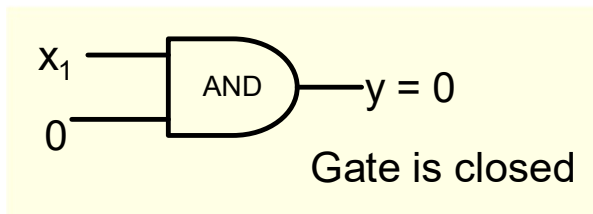
# Implementing Boolean expressions

## Elementary Gates

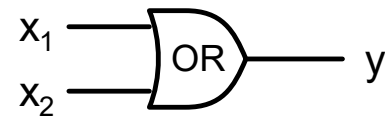
$$\text{AND: } y = x_1 \cdot x_2$$



Why call it a gate?



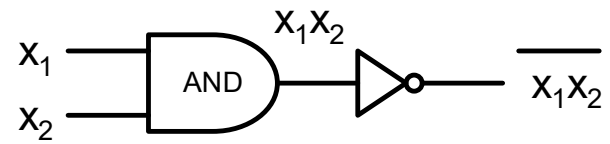
$$\text{OR: } y = x_1 + x_2$$



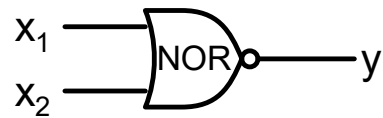
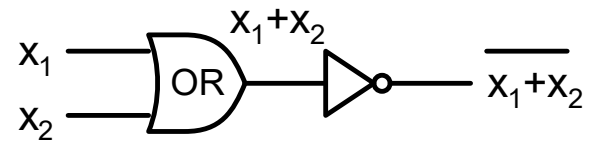
$$\text{NOT: } y = \bar{x}$$



NAND:  $y = \overline{x_1 \cdot x_2}$



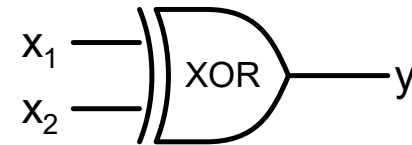
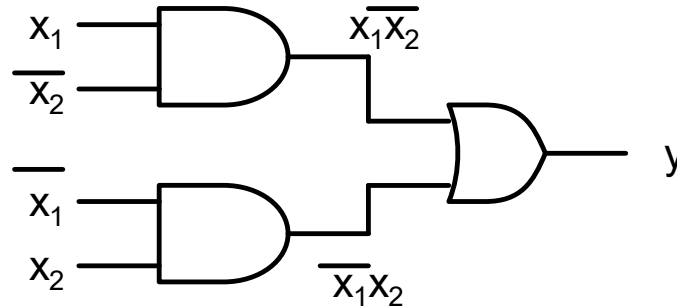
NOR:  $y = \overline{x_1 + x_2}$



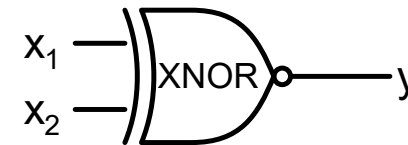
$$\text{XOR: } y = x_1 \oplus x_2 = x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2$$

Y is 1 if only one variable is 1 and the other is zero

$x_1$	$x_2$	$y$
0	0	0
0	1	1
1	0	1
1	1	0



$$\text{XNOR: } y = x_1 \odot x_2 = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2}$$

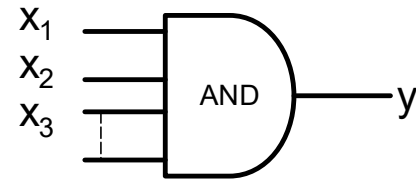


Y is 1 if only both variables are either 0 or 1

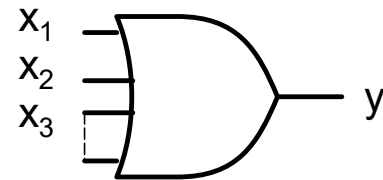
$$y = x_1 \odot x_2 = \overline{x_1 \oplus x_2}$$

## Gates with more than 2 inputs

AND:  $y = x_1 \cdot x_2 \cdot x_3 \dots$



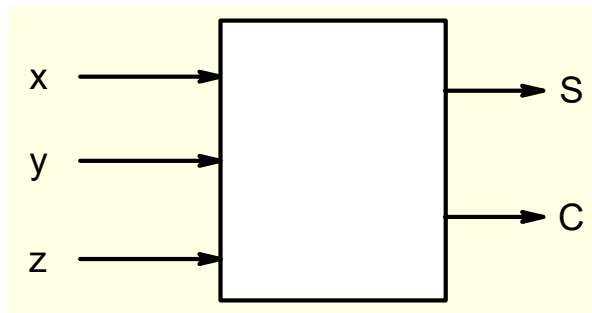
OR:  $y = x_1 + x_2 + x_3 + \dots$



XOR:  $y = x_1 \oplus x_2 \oplus x_3 = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$

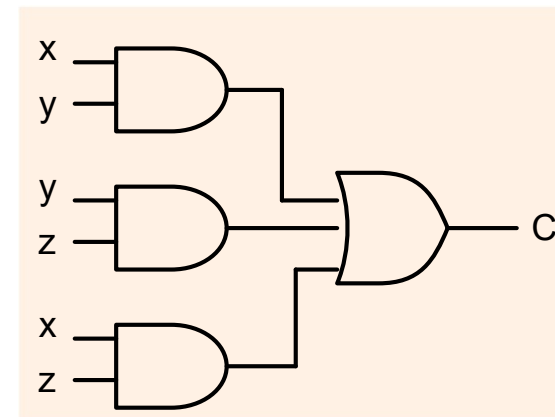
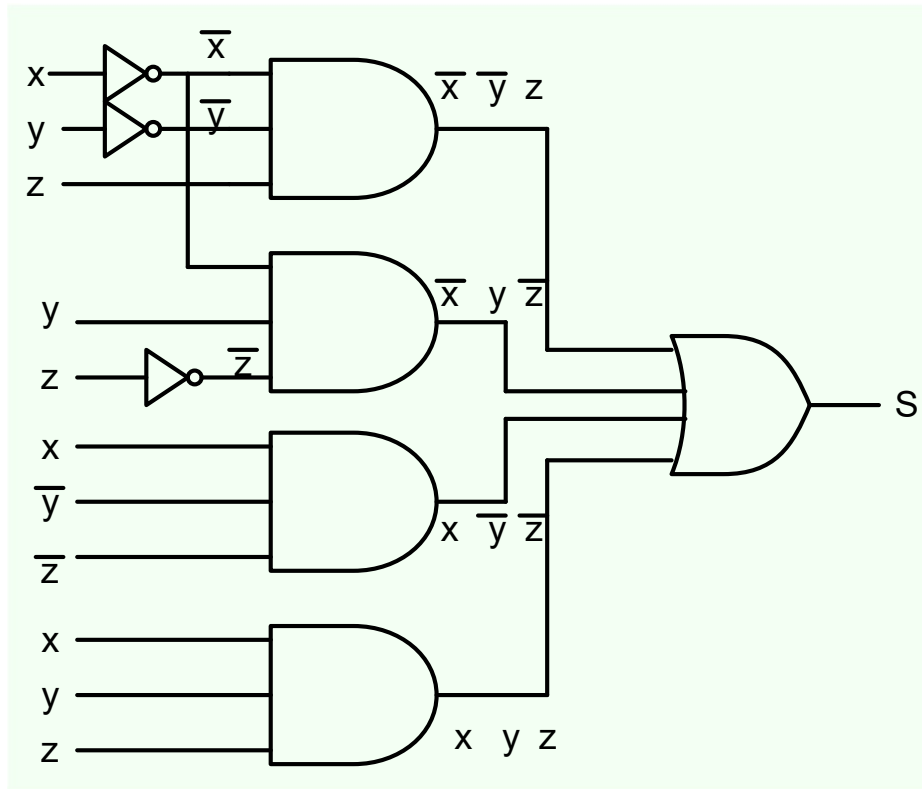
$Y = 1$  only if odd number of inputs is 1

## Implementing Boolean expressions using gates



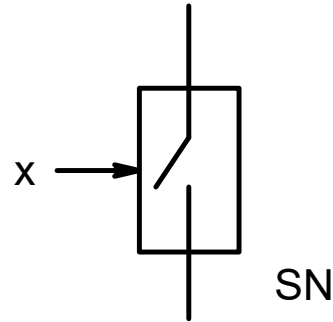
$$S = \overline{\overline{x}}.\overline{\overline{y}}.\overline{\overline{z}} + \overline{\overline{x}}.\overline{\overline{y}}.\overline{\overline{z}} + \overline{\overline{x}}.\overline{\overline{y}}.\overline{\overline{z}} + \overline{\overline{x}}.\overline{\overline{y}}.\overline{\overline{z}}$$

$$C = x.y + x.z + y.z$$



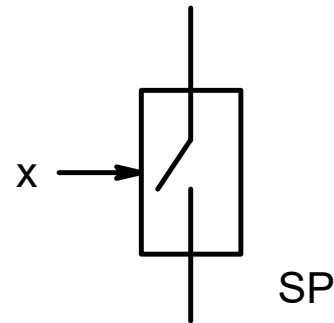
## Implementing gates using Switches

Voltage controlled Switch SN:



Switch is closed if voltage x is HIGH  
Switch is open if voltage x is LOW

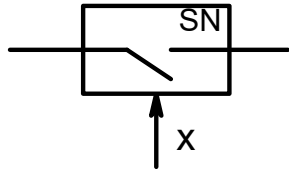
Voltage controlled Switch SP:



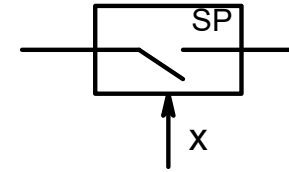
Switch is closed if voltage x is LOW  
Switch is open if voltage x is HIGH

We have seen earlier that transistors act as switches !

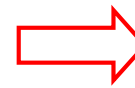
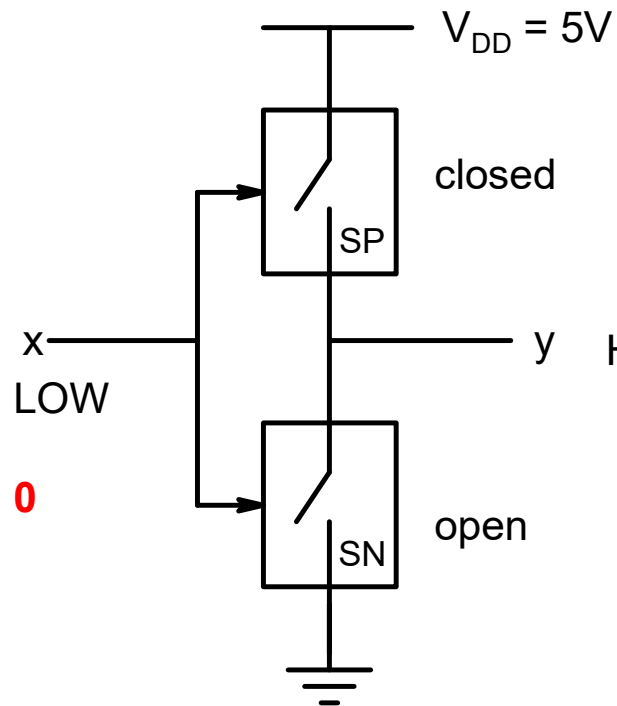




Switch is closed if voltage x is HIGH  
Switch is open if voltage x is LOW



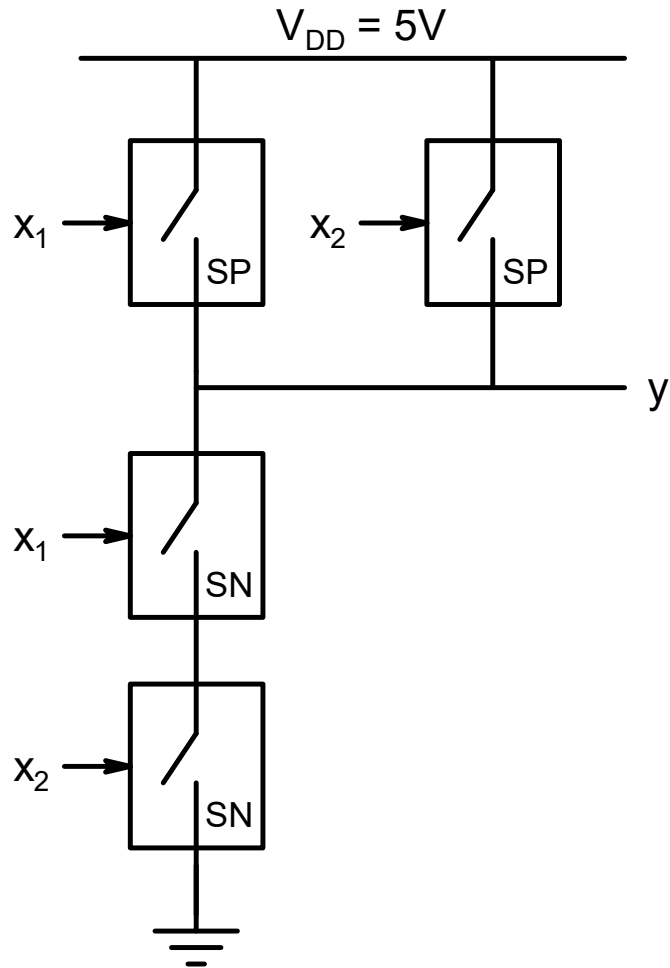
Switch is closed if voltage x is LOW  
Switch is open if voltage x is HIGH



**NOT gate**

## NAND Gate

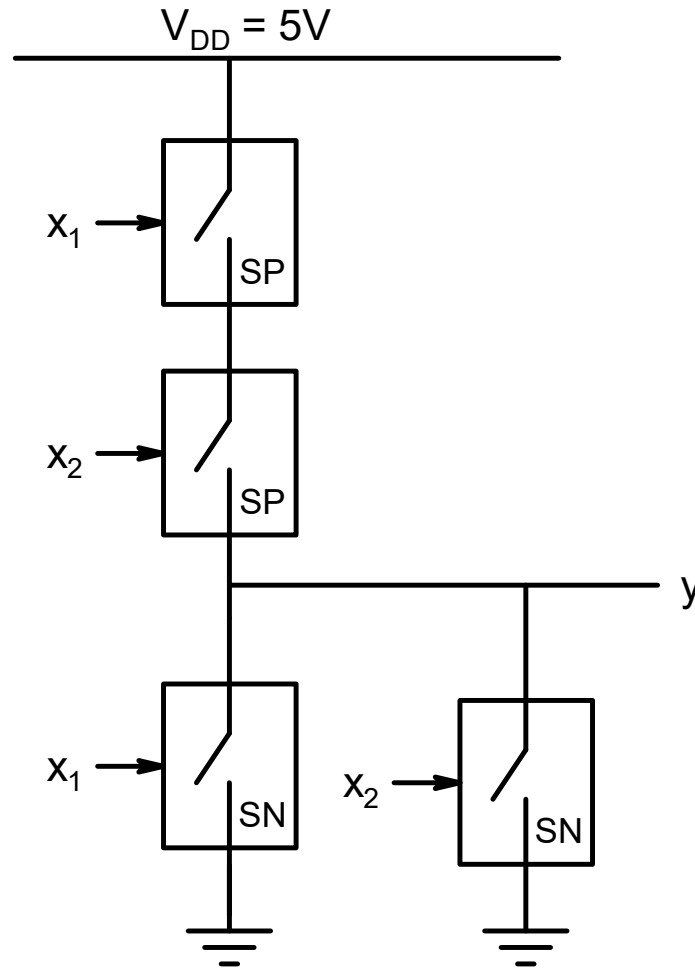
$$\text{NAND: } y = \overline{x_1 \cdot x_2}$$



$x_1$	$x_2$	$y$
LOW	LOW	HIGH
LOW	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	HIGH	LOW

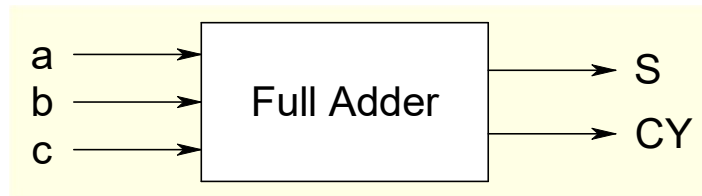
## NOR Gate

$$\text{NOR: } y = \overline{x_1 + x_2}$$



$x_1$	$x_2$	$y$
LOW	LOW	HIGH
LOW	HIGH	LOW
HIGH	LOW	LOW
HIGH	HIGH	LOW

## Design Overview

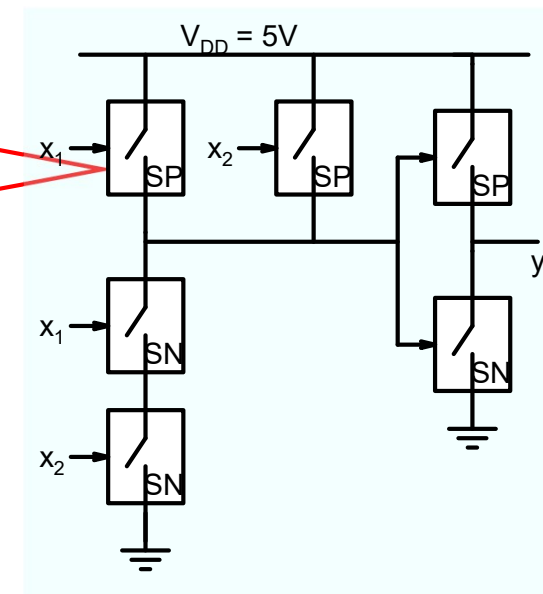
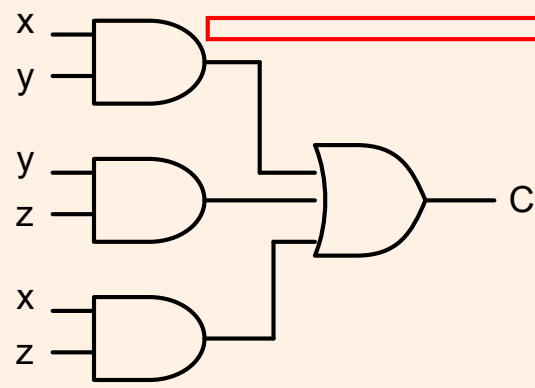
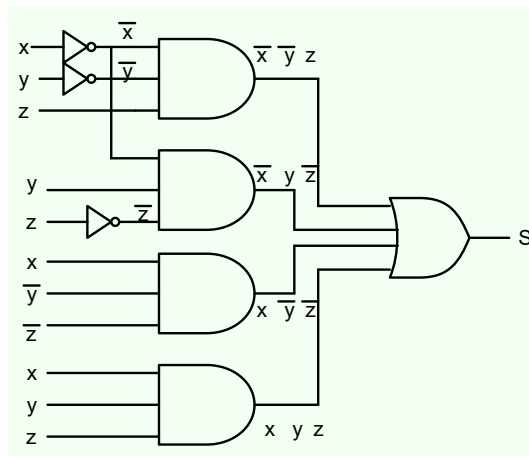


a	b	c	S	CY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



$$S = \bar{x}.\bar{y}.z + \bar{x}.y.\bar{z} + x.\bar{y}.\bar{z} + x.y.z$$

$$C = x.y + x.z + y.z$$



## Representation of Boolean Expressions

x	y	$f_1$
0	0	0
0	1	1
1	0	1
1	1	0

$$f_1 = \bar{x} \cdot y + x \cdot \bar{y}$$

x	y	min term
0	0	$\bar{x} \cdot \bar{y}$ m0
0	1	$\bar{x} \cdot \underline{y}$ m1
1	0	$x \cdot \bar{y}$ m2
1	1	$x \cdot y$ m3

$$f_1 = m_1 + m_2$$

$$f_1 = \sum (1, 2)$$

$$f_2 = \sum (0, 2, 3) = ?$$

$$f_2 = \bar{x} \cdot \bar{y} + x \cdot \bar{y} + x \cdot y$$

A minterm is a product that contains all the variables used in a function

### Three variable functions

x	y	z	min terms	
0	0	0	$\overline{x} \cdot \overline{y} \cdot \overline{z}$	m0
0	0	1	$\overline{x} \cdot \overline{y} \cdot z$	m1
0	1	0	$\overline{x} \cdot y \cdot \overline{z}$	m2
0	1	1	$\overline{x} \cdot y \cdot z$	m3
1	0	0	$x \cdot \overline{y} \cdot \overline{z}$	m4
1	0	1	$x \cdot \overline{y} \cdot z$	m5
1	1	0	$x \cdot y \cdot \overline{z}$	m6
1	1	1	$x \cdot y \cdot z$	m7

$$f_2 = \sum (1, 4, 7) = ?$$

$$f_2 = \overline{x} \cdot \overline{y} \cdot z + x \cdot \overline{y} \cdot \overline{z} + x \cdot y \cdot z$$

## Product of Sum Terms Representation

x	y	$f_1$
0	0	1
0	1	0
1	0	0
1	1	1

$$f_1 = (x + \bar{y}).(\bar{x} + y)$$

x	y	Max term
0	0	$x + \underline{y}$ M0
0	1	$\underline{x} + y$ M1
1	0	$\underline{x} + \underline{y}$ M2
1	1	$\underline{x} + y$ M3

$$f_1 = M_1.M_2$$

$$f_1 = \Pi(1, 2)$$

$$f_2 = \Pi(0, 3) = ?$$

$$f_2 = (x + y).(\bar{x} + \bar{y})$$

x	y	z	Max. terms
0	0	0	$x + y + z$ M0
0	0	1	$x + y + \bar{z}$ M1
0	1	0	$x + \bar{y} + z$ M2
0	1	1	$x + \bar{y} + \bar{z}$ M3
1	0	0	$\bar{x} + y + z$ M4
1	0	1	$\bar{x} + y + \bar{z}$ M5
1	1	0	$\bar{x} + \bar{y} + z$ M6
1	1	1	$\bar{x} + \bar{y} + \bar{z}$ M7

$$f_1 = \Pi(1, 5, 7) = ?$$

$$f_2 = (x + y + \bar{z}).(\bar{x} + y + \bar{z}).(\bar{x} + \bar{y} + \bar{z})$$

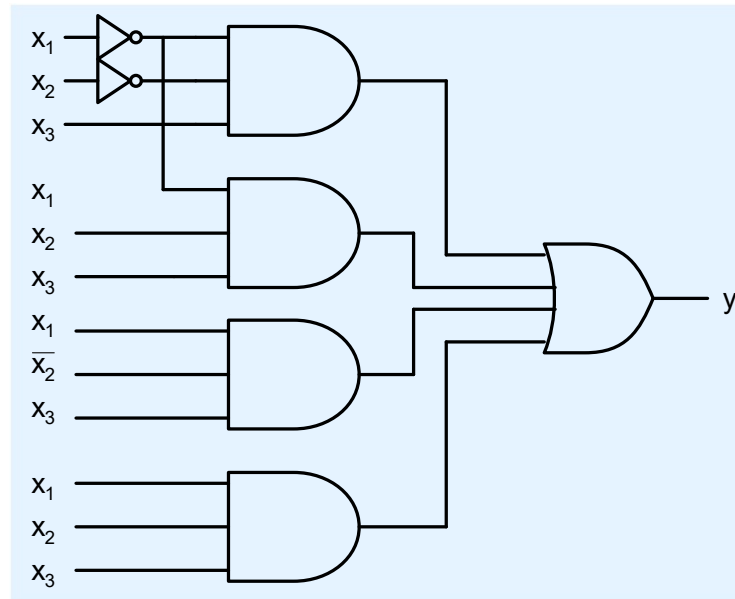


## Simplification

$x_1$	$x_2$	$x_3$	$y$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$$y = \sum (1, 3, 5, 7)$$

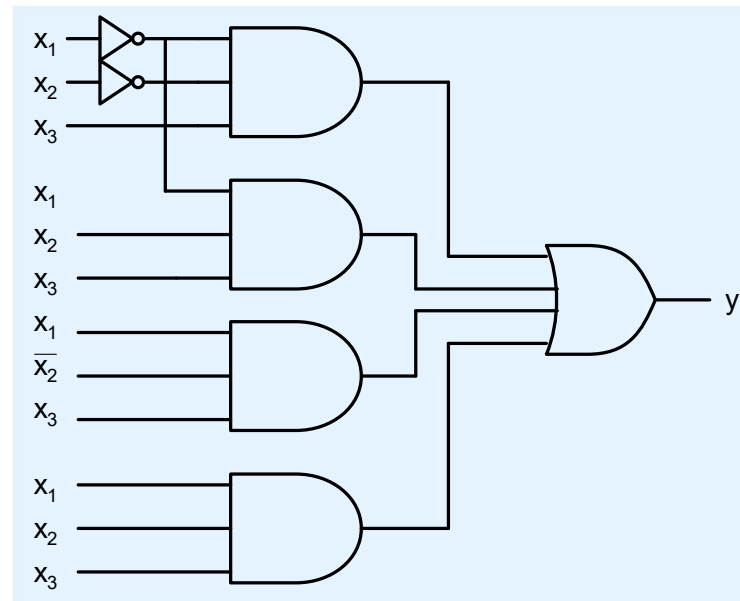
$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$



**Simplification of Boolean expression yields :  $y = x_3$  !! which does not require any gates at all !**

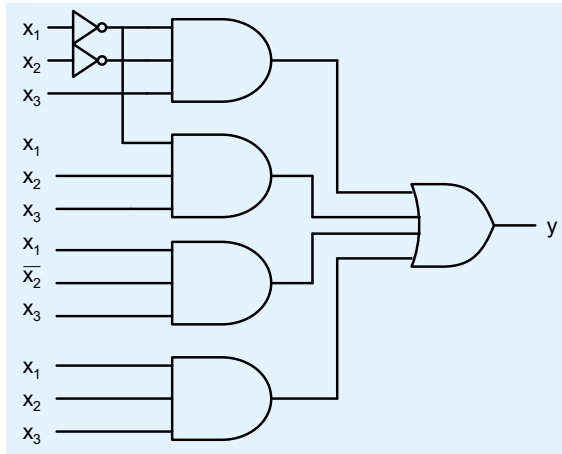
## Goal of Simplification

$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

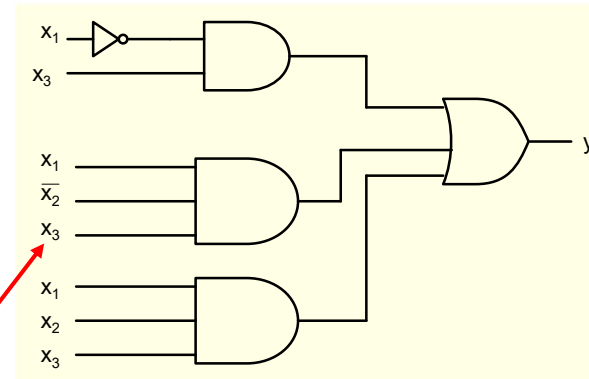


Goal of simplification is to reduce the complexity of gate circuit. This requires that we minimize the number of gates. Since number of gates depends on number of minterms, one of the goals of simplification is to **minimize the number of minterms in SOP expression**

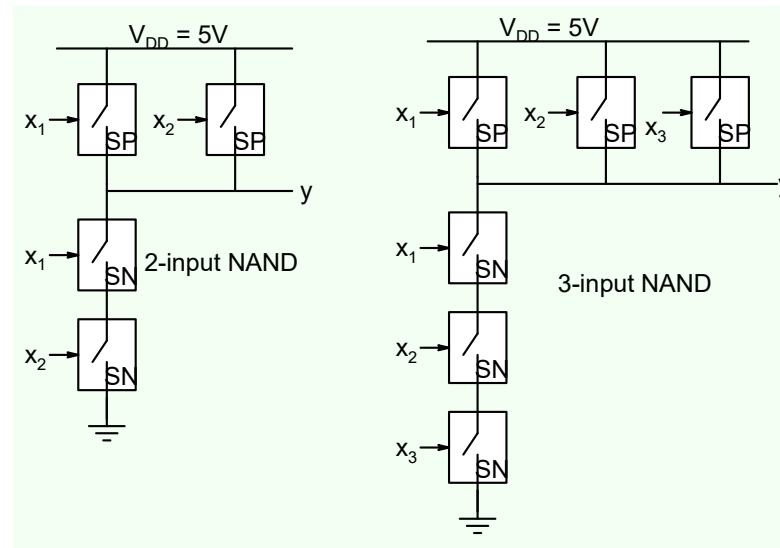
$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$



$$\Rightarrow y = \overline{x_1} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$



This circuit is simpler not just because it uses 4 gates instead of 5 but also because circuit-2 uses one 2-input and three 3-input gates as compared to five 3-input gates used in circuit-1



## Goal of Simplification

$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 \Rightarrow y = \overline{x_1} \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

In the SOP expression:

1. Minimize number of product terms
2. Minimize number of literals in each term

Simplification  $\Rightarrow$  Minimization

## Minimization

$$y = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3$$

$$y = \overline{x_1} \cdot x_3 \cdot (\overline{x_2} + x_2) + x_1 \cdot x_3 \cdot (\overline{x_2} + x_2)$$

$$y = \overline{x_1} \cdot x_3 + x_1 \cdot x_3$$

$$y = (\overline{x_1} + x_1) \cdot x_3$$

$$y = x_3$$

$$\text{Principle used: } x + \overline{x} = 1$$

$$f = \bar{x} \cdot \bar{y} + \bar{x} \cdot y + x \cdot \bar{y}$$

Apply the Principle:  $x + \bar{x} = 1$  to simplify

$$f = \bar{x} \cdot (\bar{y} + y) + x \cdot \bar{y}$$

$$f = \bar{x} + x \cdot \bar{y}$$

How do we simplify further?

$$f = \bar{x} \cdot \bar{y} + \bar{x} \cdot y + x \cdot \bar{y} = \bar{x} \cdot \bar{y} + \bar{x} \cdot \bar{y} + \bar{x} \cdot y + x \cdot \bar{y}$$

Principle used :  $x + x = x$

$$\begin{aligned} f &= \bar{x} \cdot \bar{y} + \bar{x} \cdot y + \bar{x} \cdot \bar{y} + x \cdot \bar{y} \\ &= \bar{x} \cdot (\bar{y} + y) + (\bar{x} + x) \cdot \bar{y} = \bar{x} + \bar{y} \end{aligned}$$

**Simplify**

$$f = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \\ x_1 \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} + x_1 \cdot x_2 \cdot x_3 \cdot \overline{x_4}$$

Principle:  $x + \overline{x} = 1$  and  $x + x = x$

**Need a systematic and simpler method for applying these two principles**

Karnaugh Map (K map) is a popular technique for carrying out simplification

It represents the information in problem in such a way that the two principles become easy to apply